Java Script

The Enroute Way

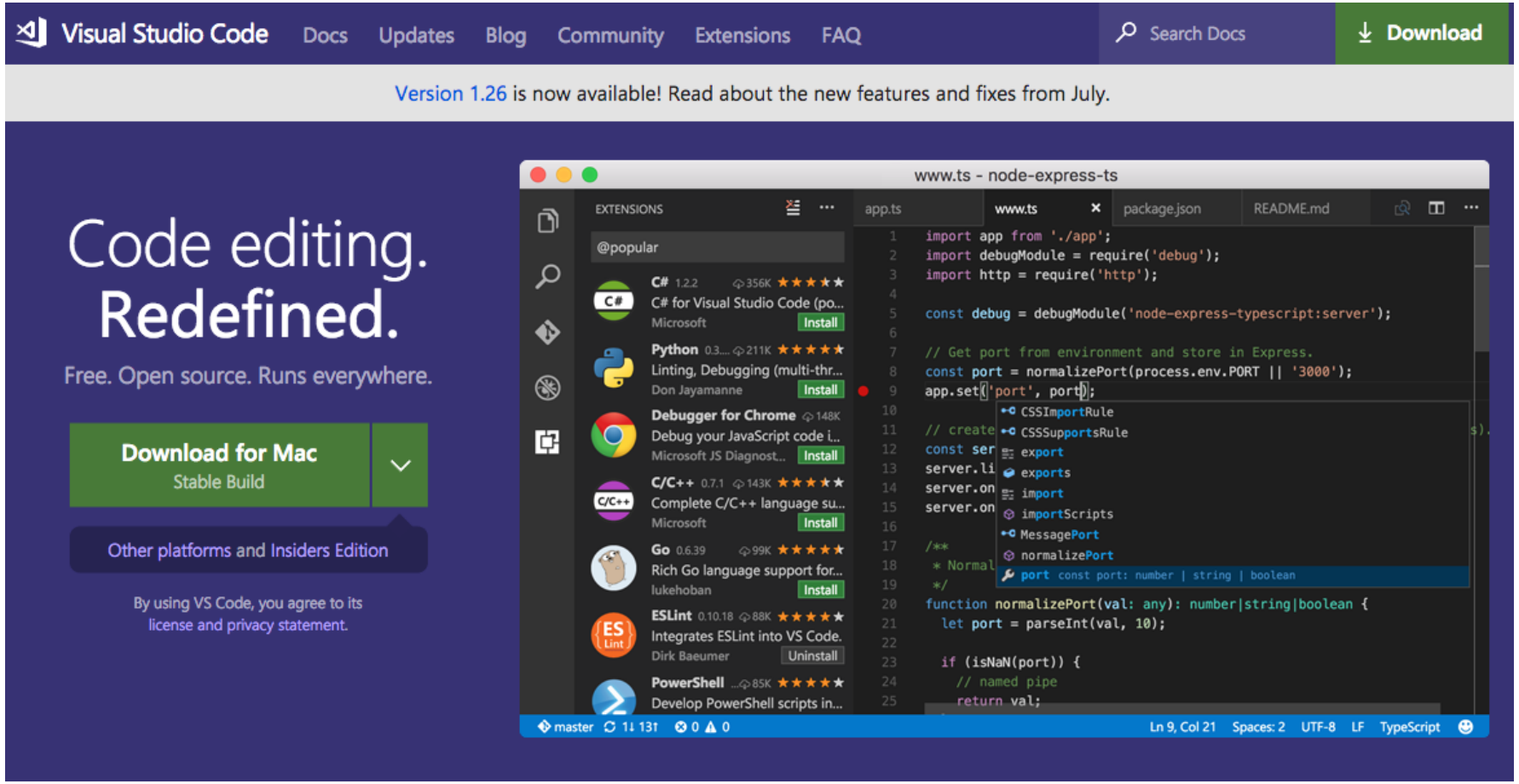
Index

- Training Setup
- HTML
- Emmet
- CSS
- Javascript Part I: Language Elements
- Javascript Part II: Functional Programming
- Javascript Part III: Advanced programming

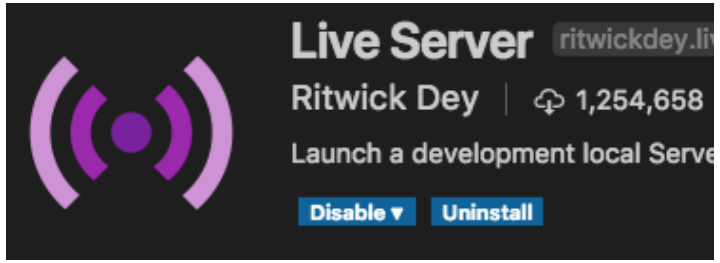
© Enroute 2017 - 2018

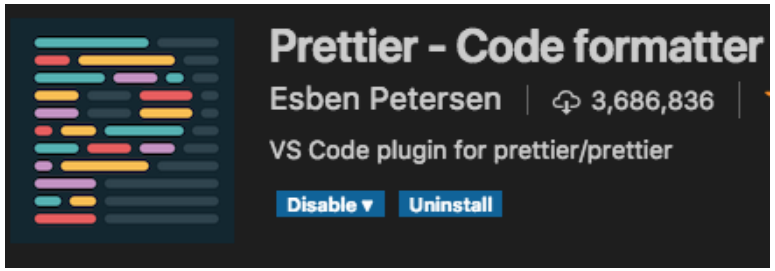
Environment Setup

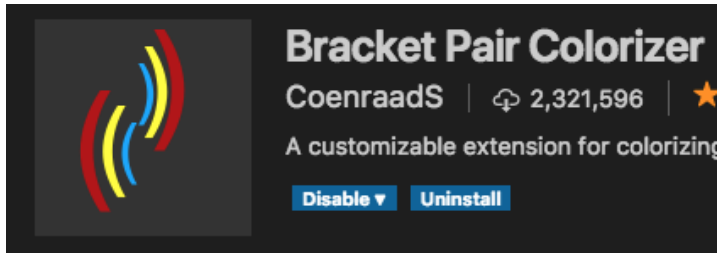
Setup – Visual Studio Code

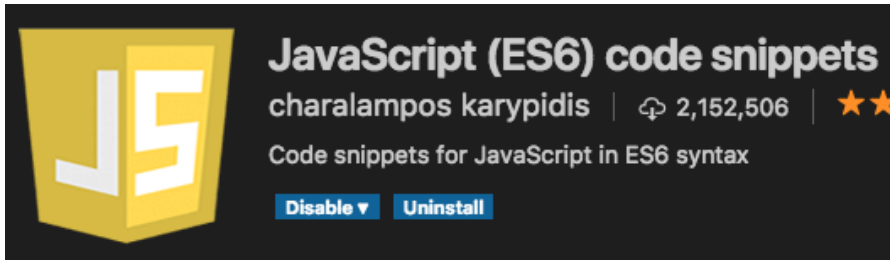


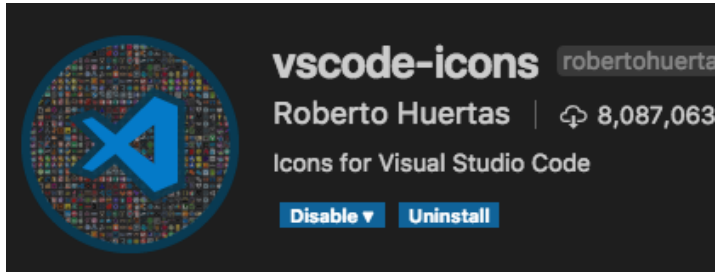
Setup – Visual Studio Code extensions











User Settings

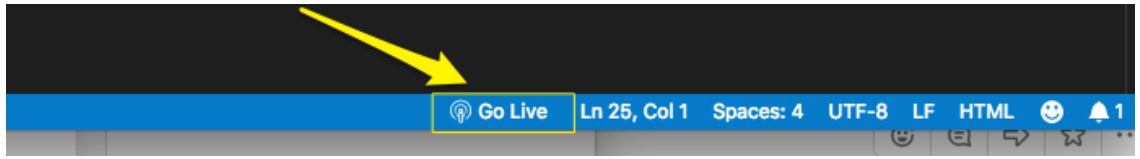
```
"editor.formatOnSave": true,
"html.format.indentHandlebars": true,
"html.format.indentInnerHtml": true,
```

© Enroute 2017 - 2018

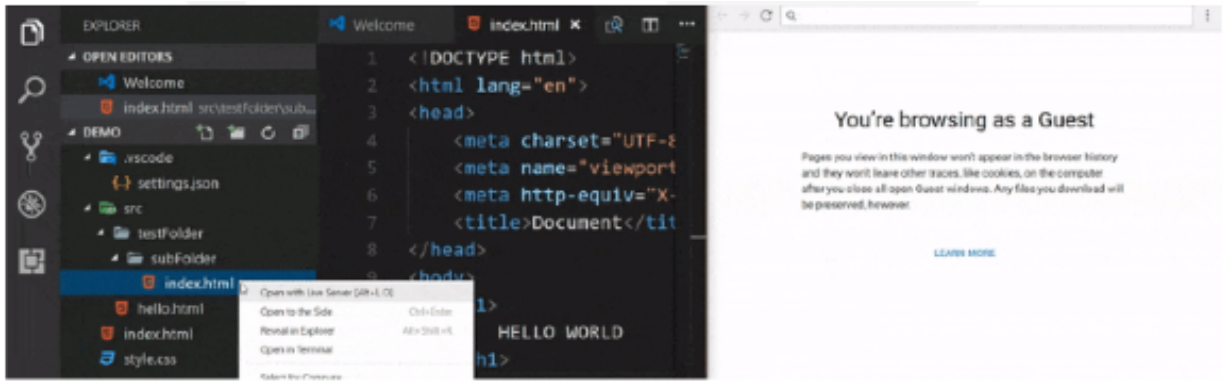
Live Server Extension

- Launch a development local Server with live reload feature

1. Open a project and directly click to Go Live from StatusBar to turn on/off the server



2. Rigth click on a HTML file and click “Open with Live Server”



Emmet Web Development toolkit

Emmet toolkit

- VS Code supports Emmet snippet expansion by dynamically parsing *CSS-like* expressions and *producing* HTML code automagically. **No extension is required!**

```
#page>div.logo+ul#navigation>li*5>a{Item $}
```

```
<div id="page">
    <div class="logo"></div>
    <ul id="navigation">
        <li><a href="">Item 1</a></li>
        <li><a href="">Item 2</a></li>
        <li><a href="">Item 3</a></li>
        <li><a href="">Item 4</a></li>
        <li><a href="">Item 5</a></li>
    </ul>
</div>
```

Emmet Syntax

- **Elements:** you can use element's names like ***div, p, header, ul, li, footer*** to *generate* HTML tags.
- **ID and Class:** emmet can be used to *add* class (#) and ID (.) attributes to specific HTML elements (tags):

```
div.header
```

→

```
<div class="header"></div>
```

```
div#footer
```

→

```
<div id="footer"></div>
```

- Custom attributes: you can use [attr] notation to add custom attributes to your elements

```
td[title="TestTittle" colspan=4]
```

→

```
<td title="TestTittle" colspan="4"></td>
```

Emmet Syntax

- **Item Numbering:** with multiplication * operator you can repeat elements, but with **$** you can *number* them

```
ul>li.item$*5
```

→

```
<ul>
    <li class="item1"></li>
    <li class="item2"></li>
    a{Click here}  "></li>
</ul>
```

- **Text:** you can use **{ }** to add text to elements

```
a{Click here}
```

→

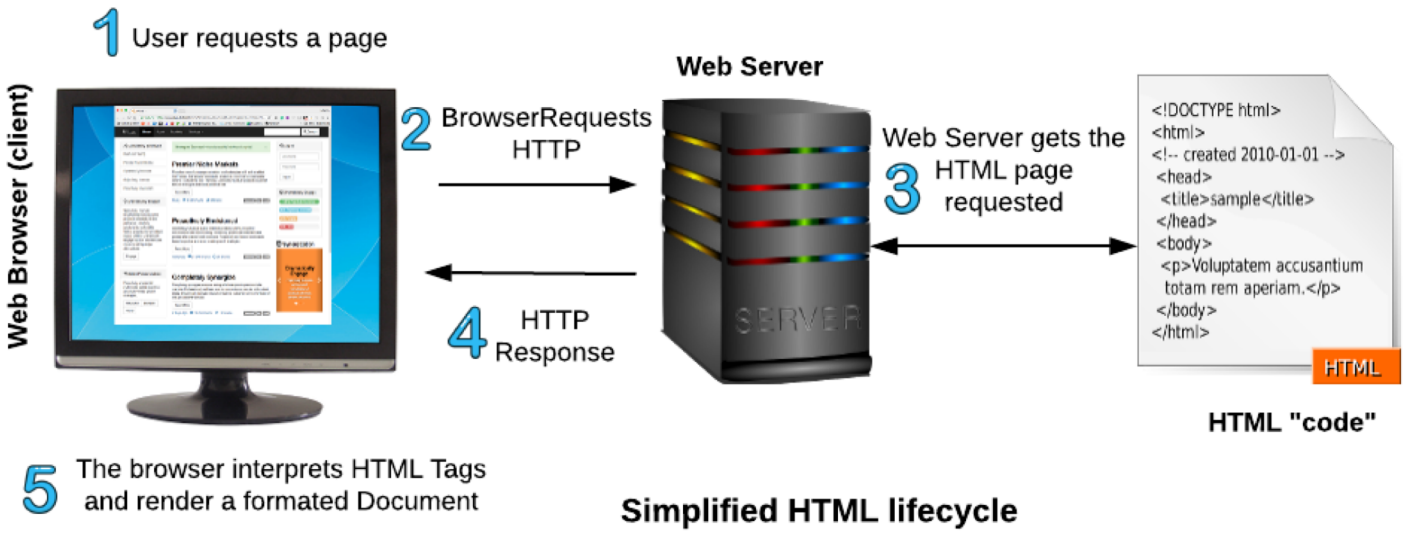
```
<a href="">Click here</a>
```

- **Emmet documentation:**
  - *https://docs.emmet.io/*
  - *https://code.visualstudio.com/blogs/2017/08/07/emmet-2.0*

Hyper Text Markup Language

HTML : Hyper Text Markup Language

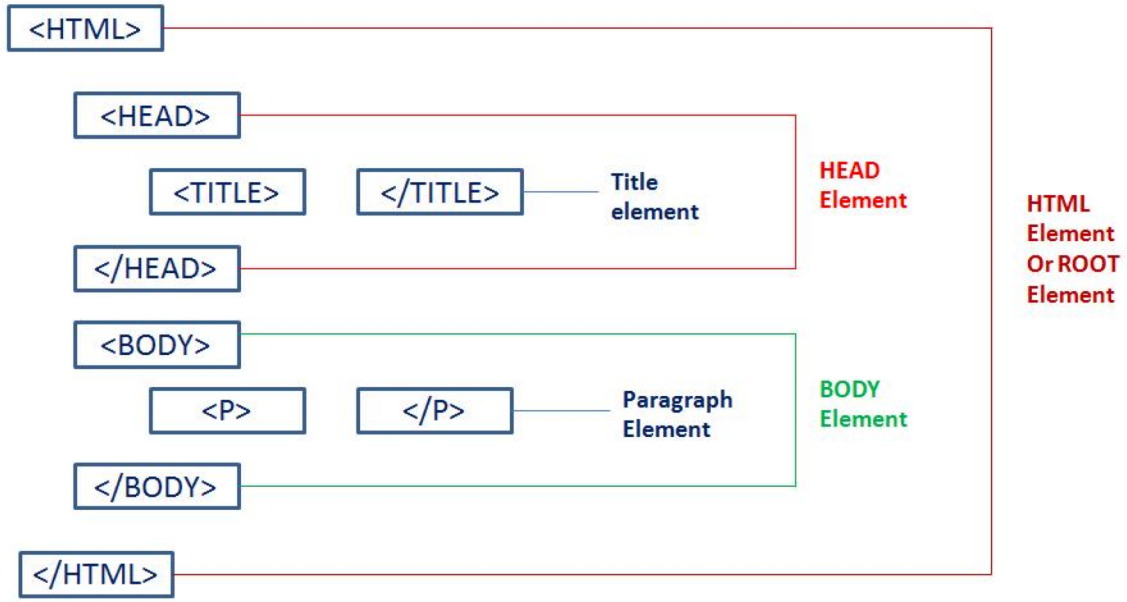
- A markup language is a computer language that uses tags to define elements within a document
- HTML Elements (tags) are “instructions” on how to format a plain document, they are the building blocks of the language
- An interpreter (browser) is needed to render the formatted document



Simplified HTML lifecycle

HTML Elements

Structure



Code

```html
<!DOCTYPE html>
<html lang="en">

    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <meta http-equiv="X-UA-Compatible" content="ie=edge">
        <title>sample</title>
    </head>

    <body>
        <p>Voluptatem accusantium totam rem aperiam.</p>
    </body>

</html>
```

© Enroute 2017 - 2018

HTML Elements and Attributes

- HTML general sintax

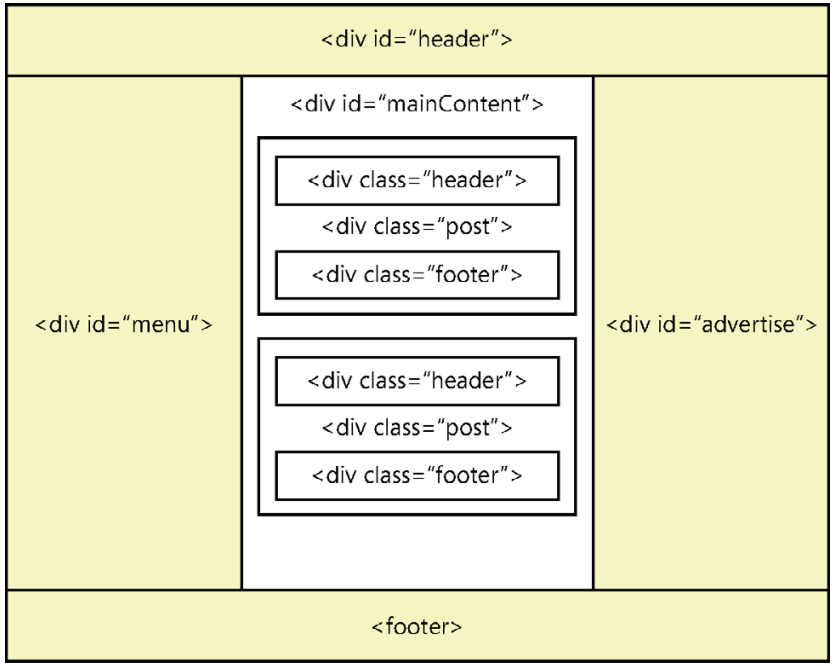
```
<tag attributeName="value">content</tag>
```

- Basic HTML Tags
  - <h1> Heading 1 </h1>
  - used to define new paragraphs
  - used to display bold text
  - italic format
  - used to define sections in the document
  - used for unordered lists:

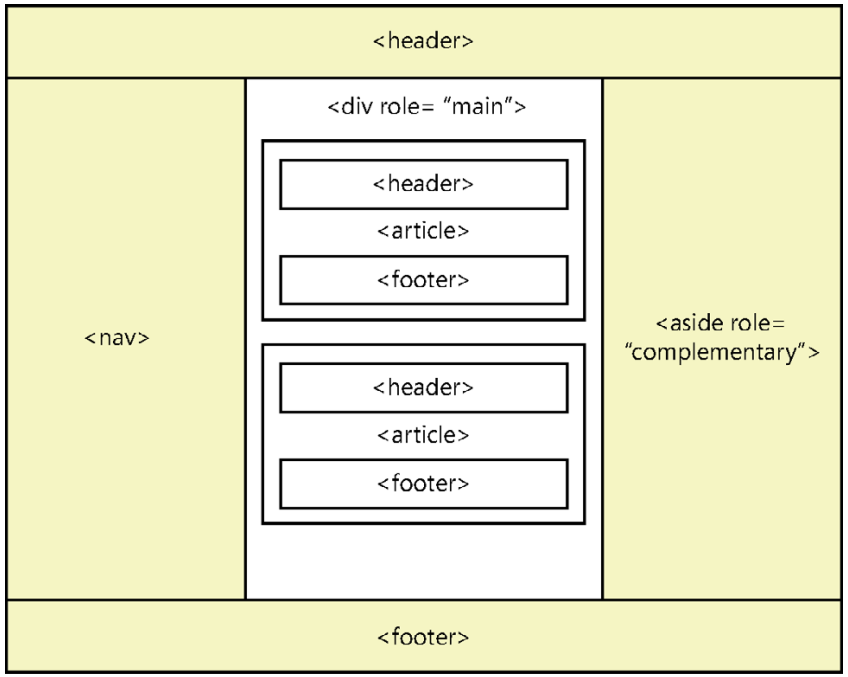
```
<ul>
    <li></li>
</ul>
```

HTML vs HTML5 layout container

- Before HTML5, * * element was used before create a layout container
- HTML5 provides a set of semantic elements in order to create a *meaningful* layout container



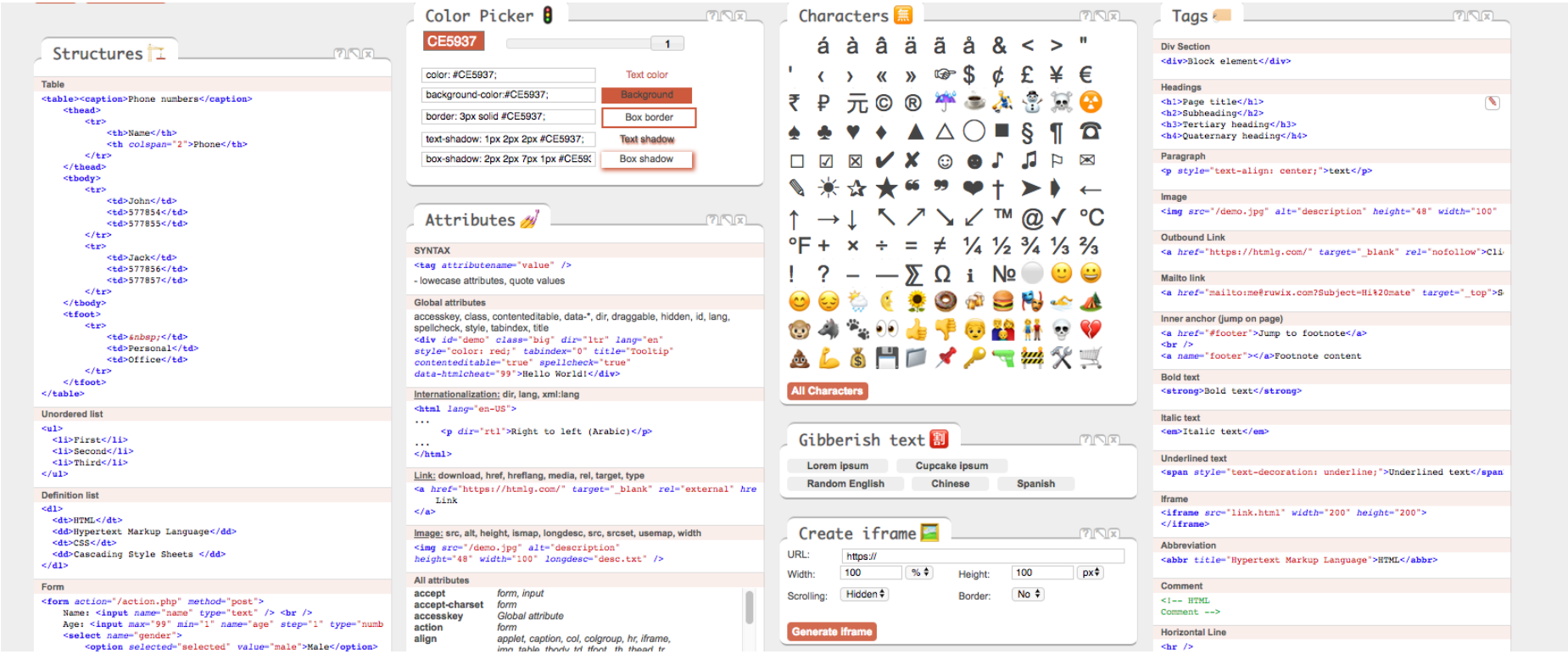
Layout container using * * elements



HTML5 Layout container

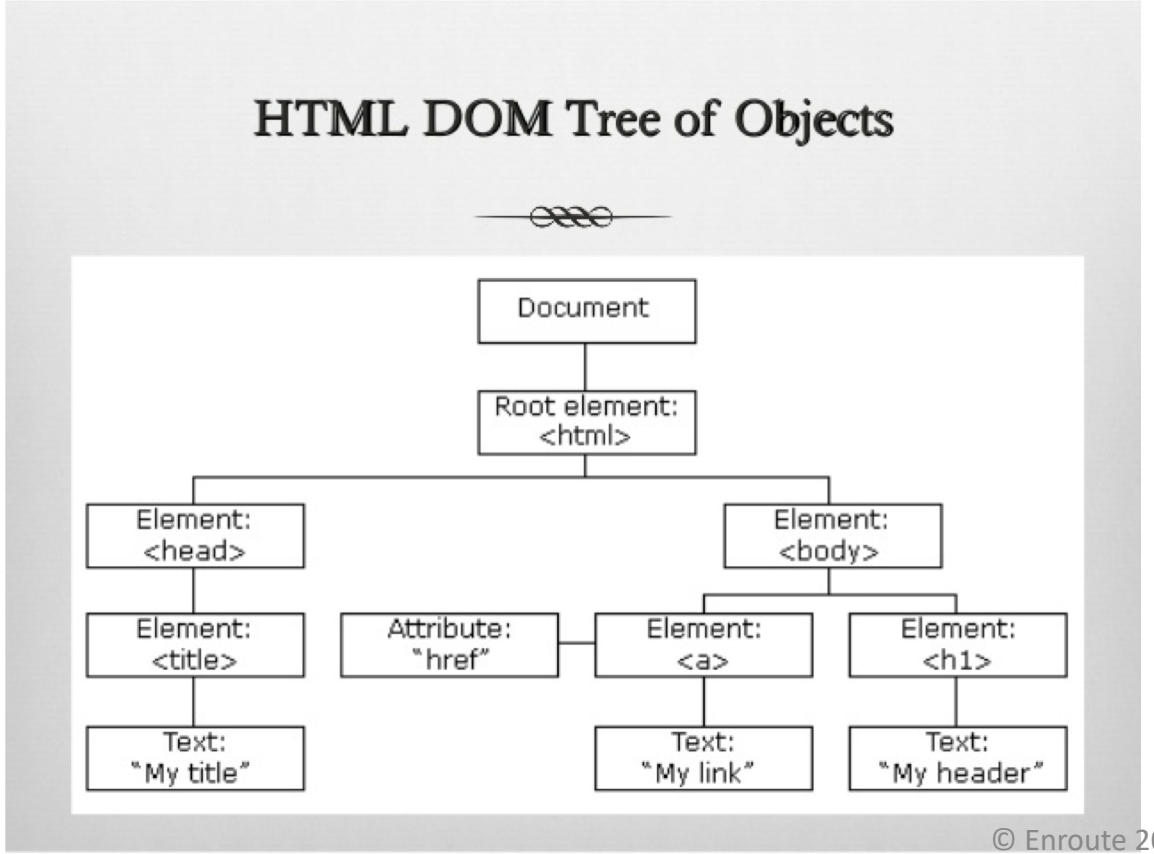
HTML online CheatSheets

- Use HTML online CheatSheet to jumpstart your HTML development skills: *https://htmlcheatsheet.com/*



HTML - DOM

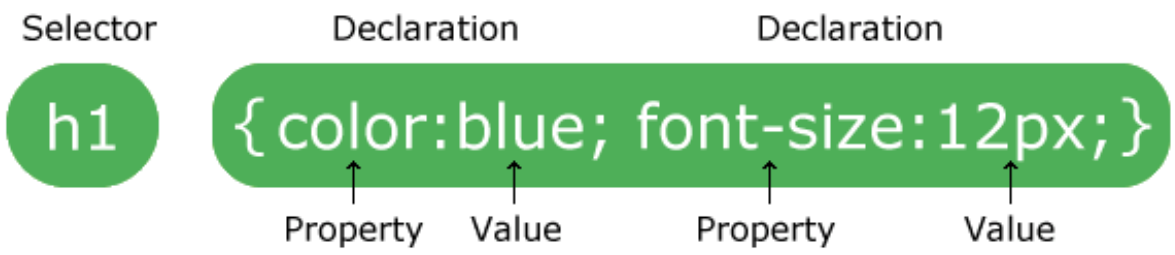
- The **D**ocument **O**bject **M**odel is the root object of HTML



CSS mini crash-curse

CSS mini-crash-curse

- CSS = **C**ascading **S**tyle **S**heets.
- Syntax: *selector + declaration block*



- CSS is a *language* that describes the style of an HTML document.
- CSS **describes** how HTML elements should be displayed
- When a browser reads a stylesheet, it formats the HTML accordingly

Three ways to add CSS to HTML:

1. Inline CSS

   Directly into html element, using *style attribute*

   style="font-family:cursive;color:blue"

2. Internal CSS

   Using <style> element in the <head> section within a single document

   <style> p {color:blue;font-size:38px;} </style>

3. External CSS (recomended!)

   Linking an external CSS file

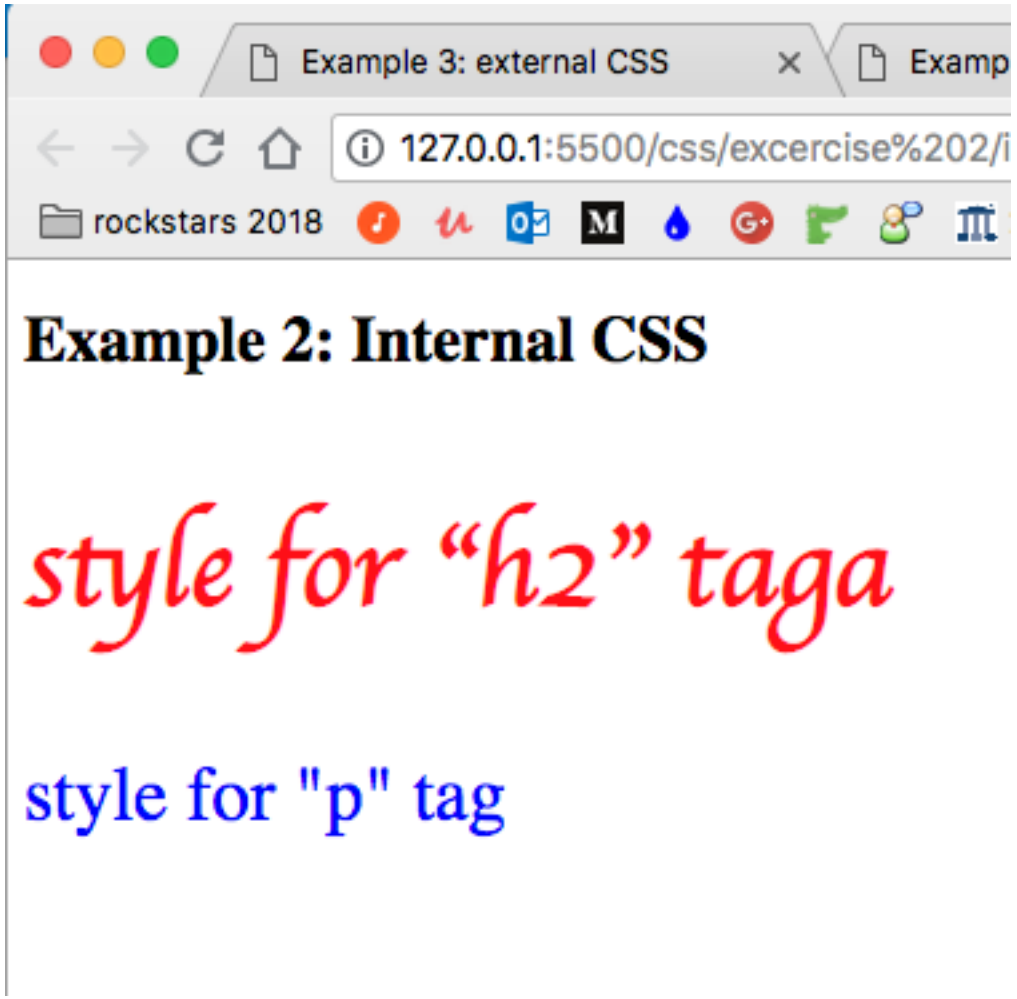
   <link rel="stylesheet" href="style.css">

Inline CSS: *style* attribute

```html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Example 1: Inline CSS</title>
  </head>
  <body>
    <h1>Example1: Inline CSS</h1>
    <h2> This paragraph has no style</h2>
    <h2 style="font-family:cursive;color: blue">This paragraph is formatted by inline CSS</h2>
  </body>
</html>
```

Internal CSS: <style> element

```html
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, in
        <meta http-equiv="X-UA-Compatible" content="ie=edge">
        <style>
            p {
                color: blue;
                font-size: 38px;
            }
            h2 {
                color: red;
                font-family: cursive;
                font-size: 58px;
            }
        </style>
        <title>Example 2: internal CSS</title>
    </head>
    <body>
        <h1>Example 2: Internal CSS</h1>
        <h2>style for "h2" taga</h2>
        <p>style for "p" tag </p>
    </body>
</html>
```



External CSS: <link> element

index.html

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-w
    <meta http-equiv="X-UA-Compatible" content="i
    <link rel="stylesheet" href="style.css">
    <title>Example 3: external CSS</title>
</head>
<body>
    <h1>Example 3: external CSS</h1>
    <h2>style for "h2" tag</h2>
    <p> style for "p" tag </p>
</body>
</html>
```

style.css

```css
p {
  color: blue;
  font-size: 38px;
}
h2 {
  color: red;
  font-family: cursive;
  font-size: 58px;
}
```

ID and **Class** Selectors

- Besides HTML Selectors (or HTML Tags), it is possible to define custom selectors as **class** and **ID** for each HTML tag

```
<div id="container">
    <p class="introduction">This is an intro paragraph</p>
    <p class="introduction">this line has the same format as the previous line</p>
</div>
```

- ID can be used to identify **one specific HTML element**
- Class can be used to identify **more than one HTML element**
- This way you can have the same HTM Tag but formatted differently depending on its class or Id

ID vs Class Selector

ID Selector

- ID selector uses the ID attribute in HTML to select a specific element

HTML: `<div id="container">`

- The hash (#) character + id is used in CSS to select the HTML element

CSS: `#container {color: purple; font-size: blue}`

Class Selector

- The class selector selects HTML elements with a specific class attribute

HTML: `<p class="intro">this is a class</p>`

- The period (.) character + class name is used to select the HTML element(s)

CSS: `.intro {color: purple; font-size: blue}`

© Enroute 2017 - 2018

ID vs Class Selector -code

index.html

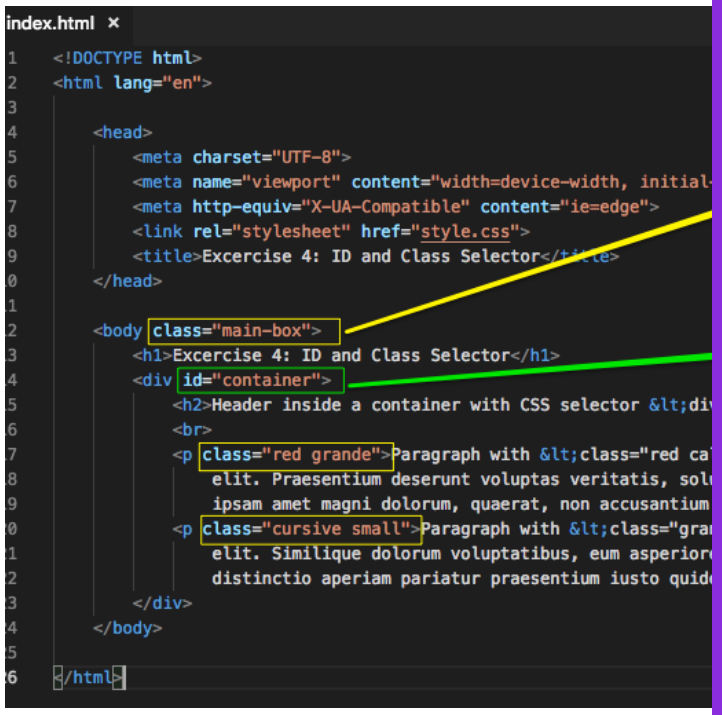
```html
<!DOCTYPE html>
<html lang="en">

    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <meta http-equiv="X-UA-Compatible" content="ie=edge">
        <link rel="stylesheet" href="style.css">
        <title>Excercise 4: ID and Class Selector</title>
    </head>

    <body class="main-box">
        <h1>Excercise 4: ID and Class Selector</h1>
        <div id="container">
            <h2>Header inside a container with CSS selector &lt;div id="container"></h2>
            <br>
            <p class="red grande">Paragraph with &lt;class="red calibri"> Lorem ipsum dolor sit amet cons
                elit. Praesentium deserunt voluptas veritatis, soluta illo aspernatur nihil fugiat porro
                ipsam amet magni dolorum, quaerat, non accusantium aut?</p>
            <p class="cursive small">Paragraph with &lt;class="grande calibri"> Lorem ipsum, dolor sit ar
                elit. Similique dolorum voluptatibus, eum asperiores harum officia delectus quo perspicia
                distinctio aperiam pariatur praesentium iusto quidem id. Minus, perferendis eius. </p>
        </div>
    </body>

</html>
```

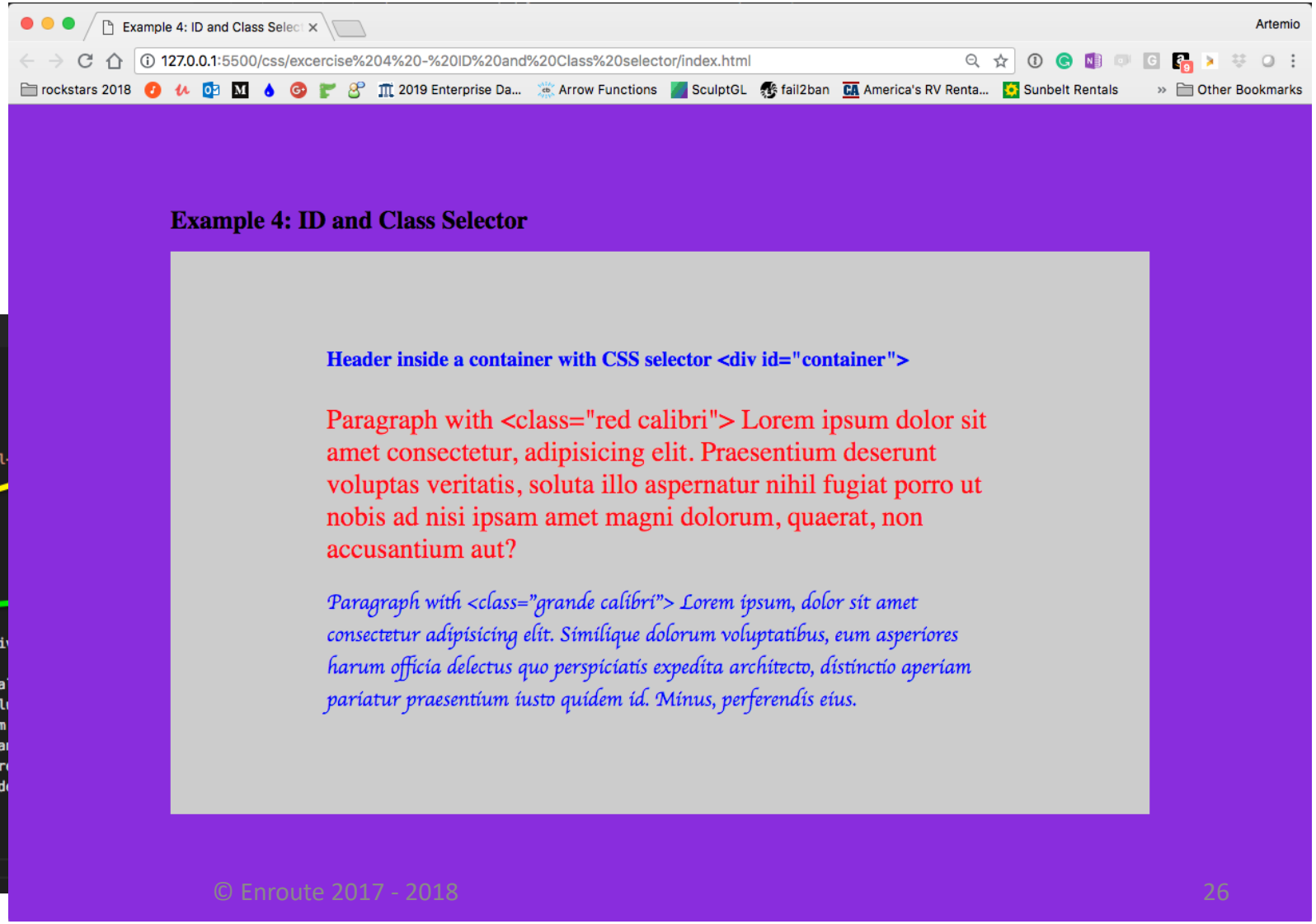
style.css

```css
.main-box {
  display: block;
  padding: 100px 200px;
  background-color: blueviolet;
}
#container {
  display: block;
  background-color: #ccc;
  color: Blue;
  padding: 100px 200px;
  font-size: 110%;
}
.red {
  color: red;
  display: inline;
}
.cursive {
  font-family: cursive;
}

.grande {
  font-size: 200%;
}

.small {
  font-size: 150%;
}
```

© Enroute 2017 - 2018

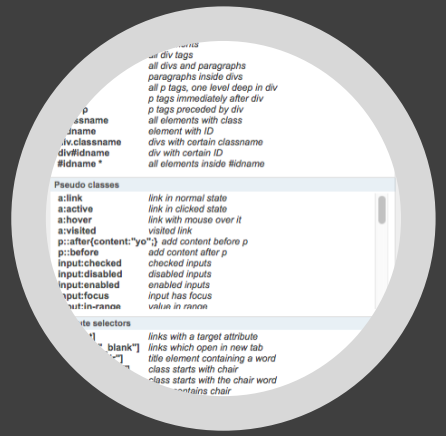
ID vs Class Selector - browser

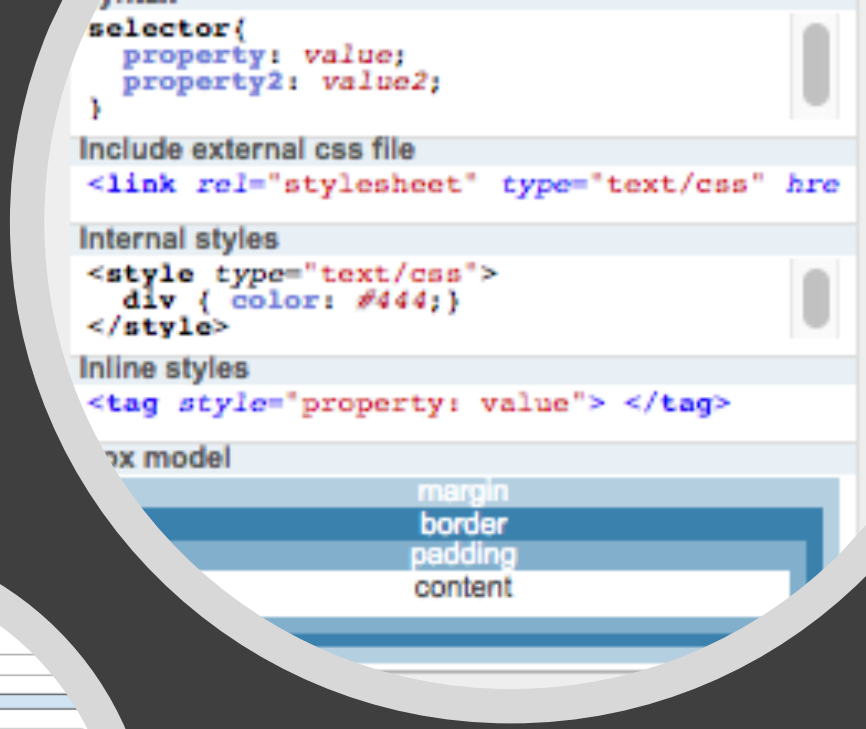


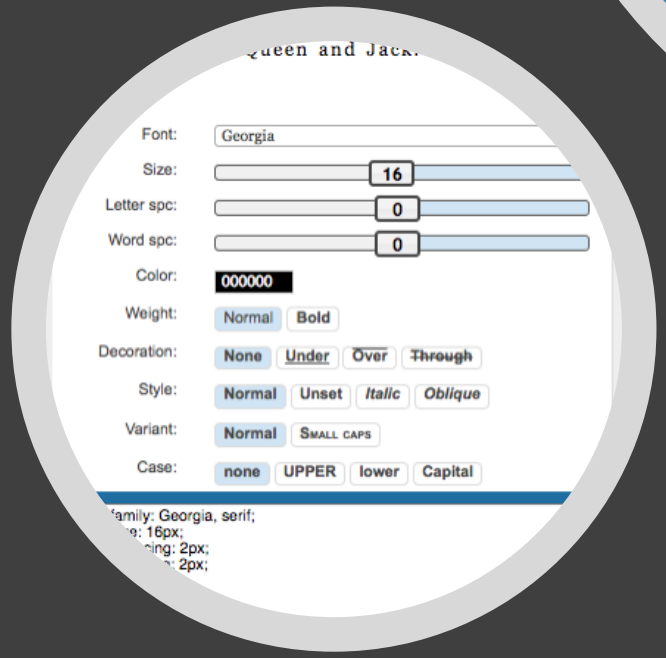


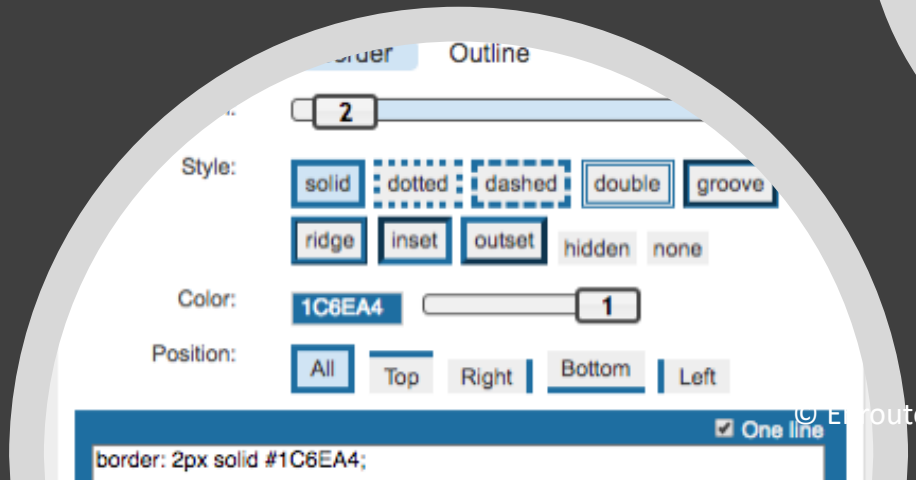
CSS cheatSheet

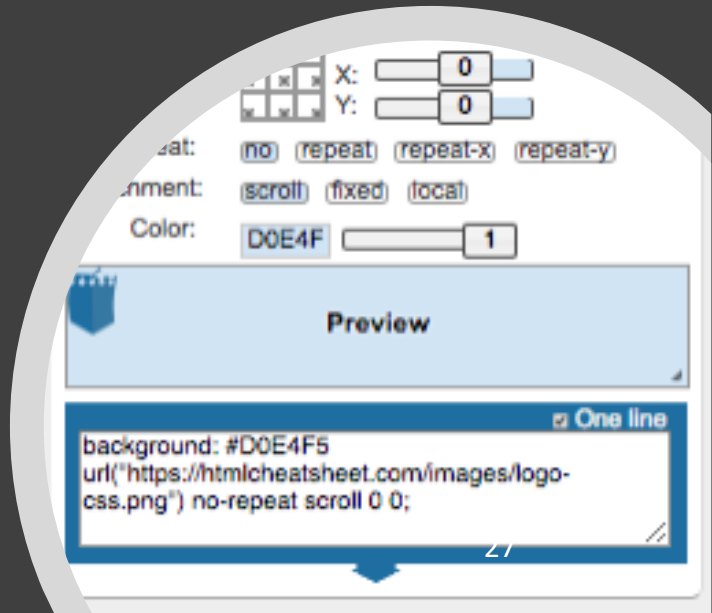
- Use CSS online CheatSheet to jumpstart your CSS development skills!
- Allows to test code snippets visually, for later to copy/paste within your pages: *https://htmlcheatsheet.com/css/*











DOM Reads

- Introduction to the DOM

  https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model/Introduction

- Document Object Model (DOM)

  https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model